3.3 enchant.js

　enchant.jsはHTML5とJavaScriptをベースとしたゲーム実行環境（エンジン）です。

enchant.jsのWebサイトから抜粋（<http://enchantjs.com/ja/>）2013.1.14

|  |
| --- |
| enchant.js is …  * カンタンにゲームやアプリを開発できる HTML5 + JavaScript フレームワークです。 * 2011年に公開され、すでに[1,000 本以上のゲーム/アプリが公開](http://enchantjs.com/examples)されています。 * オープンソース (MITライセンス) で、無料で利用できます。 * [ドキュメント](http://enchantjs.com/ja/resource-ja/api-documentation/)・[書籍](http://enchantjs.com/ja/resource-ja/books/)・[チュートリアルサイト](http://enchantjs.com/ja/for-beginners/how-to-start)が充実しています。 * たくさんの[プラグインで機能を拡張](http://enchantjs.com/plugins/)できます。 * UEI/ARC を中心としたメンバによって開発・メンテナンスされています。 * [プログラミング教育](http://enchantjs.com/educational-action)のためにも利用されています。  Features  * **オブジェクト指向**: 表示されているものはすべてオブジェクトです。 * **マルチプラットフォーム**: iOS, Android, Mac, Windows のブラウザで動作します。 * **Windows 8**: Windows 8 対応のHTML5アプリケーションを開発できます * **イベント駆動**: イベントリスナによる非同期処理を基本としています。 * **アニメーションエンジン**: 標準でトゥイーンなどのアニメーションを利用できます。 * **ハイブリッド描画**: Canvas API, DOM による描画に両対応しています。 * **WebGL 対応**: プラグインにより WebGL を使った3Dゲームに対応しています。 * **画像素材も提供**: ゲーム用画像素材も無償提供されています。 |

　enchant.jsではアクションゲーム、ノベルゲーム、RPG、アドベンチャーゲームなどの様々なゲーム開発にそれぞれ役立つプラグインが提供されています。

　まずはenchantjs.comのサイトを見て、enchant.jsを用いるとどんなものが作れるのか調べてみてください。

|  |  |  |
| --- | --- | --- |
| **【演習：初めての開発（enchant.js）】**  enchanto.jaを使ったアプリはローカルでも開発して実行できますが、この演習では手軽に使えるオンラインエディタでアプリケーションを開発・動作確認してみましょう。  1. オンライエディタのサイトにアクセス・ログインをする  (1)以下のオンラインエディタのサイトにWebブラウザでアクセスする  http://jsdo.it/  (2)「Start coding」のボタンをクリック  (3)「GitHub IDでログイン」を選ぶ  ※ブラウザでGitHubにログイン中であれば「Authorize app」で使用開始できます。  　ログインしていない人はアカウントとパスワードを入力してログインしてください。  (4)名前を設定して「submit」する  ※ソースコードの編集画面が表示されます  2. ソースコード編集画面を確認する  タイトル編集領域    切り替えタブ  プレビュー領域  ソースコード編集領域  3. タイトルを設定する  　タイトルや説明文などの基本情報が編集できるようになります  (1)タイトル編集領域をクリックし、「Title」欄に「はじめてのenchant.js」と入力  4. enchant.jsライブラリの追加  　JavaScriptにenchant.jsの機能（パッケージ）を追加します  (1)「ソースコード編集領域」の上部にある「切り替えタブ」の「JavaScript」をクリック  (2)「＋ Add Library」をクリック  (3)「Major Library」の欄で「enchant.js」を選んで「Add」ボタンをクリック  5. ソースコード（JavaScript）の入力  (1)JavaScriptのソースコード領域に以下のコード（スクリプト）を入力   |  | | --- | | enchant(); //…①  var picNumber = 0; //…②  window.onload = function() { //…③  var game = new Game(320, 480); //…④    game.preload('http://enchantjs.com/assets/images/chara1.gif'); //…⑤    game.onload = function() { //…⑥  var bear = new Sprite(32, 32); //…⑦  bear.image = //…⑧  game.assets['http://enchantjs.com/assets/images/chara1.gif'];  bear.frame = 0; //…⑨    bear.addEventListener(Event.ENTER\_FRAME, function() { //…⑩  picNumber++; //…⑪  if (picNumber > 3) { //…⑫  picNumber = 0;  }  bear.frame = picNumber; //…⑬  bear.x += 2; //…⑭  });    game.rootScene.addChild(bear); //…⑮  };    game.start(); //…⑯  }; |   6. HTMLの入力  　コード（JaveScript）が実行されるページの情報をHTMLで設定します  (1)「切り替えタブ」の「HTML」をクリック  (2)以下のHTMLを入力する   |  | | --- | | <!DOCTYPE html>  <html>  <head>  <meta charset="utf-8">  <title>初めてのenchant.jsアプリ</title>  <style type="text/css">  body { margin: 0; }  </style>  </head>  <body>  </body>  </html> |   7. CSSを編集する  　このアプリケーションではCSSを使わないのですべて削除します  (1)「切り替えタブ」の「CSS」をクリック  (2)デフォルトで入力されているコードをすべて削除する  8. 動作確認する  (1)「Save」ボタンを押す（保存してプレビューが行われる）  ※プレビュー領域で動作確認できます。クマが左から右に走り去れば動作成功です。  9. 編集終了する  (1)「Finish Editing」をクリック  ※再度編集するときは一覧画面のページを選び、下の方にある「Edit」をクリックする |

【ソースコード（スクリプト）の解説】

①…enchant.jsのライブラリを使用するために必要な命令です（必須）

②…このプログラムで使用する変数picNumberを定義しています。初期値は0です

③…端末画面が表示されたとき（「windowオブジェクト」が読み込み（ロード、load）されたとき）に行う処理を指定しています。既に定義されている処理を指定することができます。このスクリプトのようにこの場所に直接 『 =function() { }; 』といった形で処理内容を定義することもできます

④…gameという変数名で参照（使用）できる「Gameオブジェクト」を作成します。引数として与えた320,480の値は、画面サイズになります。

⑤…「Game」に使用する画像ファイルをあらかじめ読み込み（プレロード）させています。読み込まれた画像ファイルはアセット領域（asstes）に格納されます。引数として与えられている 'http://enchantjs.com/assets/images/chara1.gif' というURLは、enchant.jsの開発団体UEI（ユビキタスエンターテインメント）によって提供されている画像のURLです。自分のWebサイト等で公開している画像を使うこともできます

⑥…ゲーム（game）が実行された時（プレロードが完了した後に実行開始されます）に行う処理を指定しています。こちらも『 = function() { }; 』の形でこの場所に直接定義したうえで指定しています。

⑦…bearという変数名でスプライト（Sprite）を定義しています。スプライトとはゲームキャラクターの様に移動などを行わせたい、画像を持つオブジェクトのことです。引数の32, 32がスプライト画像のサイズになります

⑧…bearというスプライトに使用する画像の元となる画像ファイルを指定しています。このスクリプトでは以下のプレロードした画像を使用するよう指定しています

　　　　　　　http://enchantjs.com/assets/images/chara1.gif



⑨…画像ファイルの中のどの部分をスプライトの画像とするか指定（フレームの指定）をしています。画像ファイルの左上から右側に順番にフレーム番号が割り当てられますので上の画像だと、「直立している茶色のクマ」が番号0で、「スケボーに乗っているクマ」が番号4、「直立している白いクマ」が番号5、「直立している女の子のクマ」が番号10になります

⑩…bearというスプライトが一定時間ごとに行う処理を指定しています。これも③や⑥と同様に『 =function() { }; 』という形で処理内容をこの場所で定義しています

⑪…変数picNumberの値を1増やしています

⑫…picNumberの値を調べて3を超えていた（4になっていた）場合は0に戻しています

⑬…bearというスプライトの画像を切り替えています。bear.frameに値（フレーム番号）を設定することで、その値の位置にある画像を使用するように指定できます

⑭…bearというスプライトのX座標位置を2増やして（2ドット右移動）います。

⑮…ゲーム画面（game）のシーン（rootScene）の下位要素（child）としてbearというスプライトを追加しています

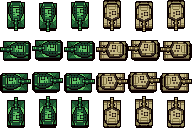
⑯…ゲーム（game）を開始させる呼び出し（スイッチ）です

応用練習

　・戦車を揺らし（画像切り替え）ながら右方向に移動するように変更しましょう。

　　画像のURLは以下のものが使用できます

　　　　　　http://enchantjs.com/assets/images/chara3.gif



　・右から左への移動に変えてみましょう

　・複数の戦車（スプライト）を同時に移動させてみましょう

■オブジェクトのプロパティ

　オブジェクトに固有の値を与えることができます。それをプロパティ（property、属性）とよびます。

　プロパティにはオブジェクトの種類ごとにあらかじめ用意されている用途を持ったものがあります。またそれ以外にもプログラマがプログラミングしやすいように各オブジェクトに自由にプロパティを設定することもできます

例） var mySprite = new Sprite(32, 32); //オブジェクトを作成

mySprite.dx = 2; // dxという属性を設定

mySprite.x += mySprite.dx; // 属性xに属性dxの値を加算

■スプライト（Spriteオブジェクト）の操作

　以下のプロパティの値を変更する事でスプライトの画像位置を変えたり、画像を変形させたりすることができます

|  |  |
| --- | --- |
| プロパティ | 役割 |
| x | X座標位置 |
| y | Y座標位置 |
| flame | フレーム番号 |
| image | 表示画像 |
| rotation | 回転角度。0～360(度)で一回転。初期値は0 |
| scaleX | X方向の倍率。1で元のサイズ。2で2倍。-1で左右反転 |
| scaleY | Y方向の倍率。1で元のサイズ。2で2倍。-1で上下表示 |

例） var mySprite = new Sprite(32, 32);

mySprite.x = 100; // (100, 120)の

mySprite.y = 120 // 場所に表示する

mySprite.scaleX = -1; // 左右反転した画像にする

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| 直接プロパティの値を変更するのではなく、以下のメソッド（関数）を使用することもできます。   |  |  | | --- | --- | | メソッド | 処理内容 | | rotate( 角度 ) | 回転 | | scale( X方向の倍率, Y方向の倍率 ) | 倍率変更 |   例） mySprite.rotate( 90 ); // 90度時計回りに回転させる |

■イベント処理

　オブジェクトがマウス操作等のイベント（出来事）を取得できるようにするには、以下のように記述します。

　　　オブジェクト.addEventListener(イベントの種類, 実行する処理);

例）

|  |
| --- |
| var mySprite = new Sprite(32, 32);  mySprite.dx = 2;  mySprite.addEventListener(Event.TOUCH\_START, function(event) {  mySprite.dx += 2;  }); |

※スプライトmySpriteをクリックするたびに属性dxの値が2増えます

イベントの種類には以下のようなものがあります

|  |  |
| --- | --- |
| イベントの種類 | イベント内容 |
| Event.ENTER\_FRAME | フレーム開始（画像表示開始） |
| Event.TOUCH\_START | マウスなどでのタッチ開始 |
| Event.TOUCH\_END | マウスなどでのタッチ終了 |

　　他のイベントやライブラリを追加することで処理可能になるイベントもあります

応用練習

　・スプライトをタッチ（マウスでクリック）すると移動速度が上がるようにしましょう

　・スプライトをタッチすると画像が変わるようにしてみましょう

　・スプライトをタッチすると移動方向が右→下→左→上に変わるようにしてみましょう

　・その他、スプライトの働きについてWeb等で調べてみましょう